IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 7, 2021, accepted January 8, 2022, date of publication January 13, 2022, date of current version February 3, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3142888

Magnetic Force Classifier: A Novel Method for Big Data Classification

AHMAD B. HASSANAT[1], **(Member, IEEE), HASAN N. ALI**[1], **AHMAD S. TARAWNEH**[2], **MALEK ALRASHIDI**[3], **MANSOOR ALGHAMDI**[3], **GHADA AWAD ALTARAWNEH**[4], **AND MOHAMMAD ALI ABBADI**[1]

[1]Faculty of Information Technology, Mutah University, Karak 61710, Jordan
[2]Department of Algorithms and Their Applications, Eötvös Loránd University, 1053 Budapest, Hungary
[3]Computer Science Department, Applied College, University of Tabuk, Tabuk 71491, Saudi Arabia
[4]Department of Accounting, Mutah University, Karak 61710, Jordan

Corresponding author: Ahmad S. Tarawneh (ahmad.trwh@gmail.com)

ABSTRACT There are a plethora of invented classifiers in Machine learning literature, however, there is no optimal classifier in terms of accuracy and time taken to build the trained model, especially with the tremendous development and growth of Big data. Hence, there is still room for improvement. In this paper, we propose a new classification method that is based on the well-known magnetic force. Based on the number of points belonging to a specific class/magnet, the proposed magnetic force (MF) classifier calculates the magnetic force at each discrete point in the feature space. Unknown examples are classified using the magnetic forces recorded in the trained model by various magnets/classes. When compared to existing classifiers, the proposed MF classifier achieves comparable classification accuracy, according to the experimental results utilizing 28 different datasets. More importantly, we found that the proposed MF classifier is significantly faster than all other classifiers tested, particularly when applied to Big datasets and hence could be a viable option for structured Big data classification with some optimization.

INDEX TERMS Artificial intelligence, classification algorithms, data mining, supervised learning, machine learning.

I. INTRODUCTION

Artificial intelligence and machine learning in particular is a hot research subarea because of their numerous applications in various fields and contexts. Examples of applications include, but are not limited to: Natural Language Processing [1]–[6], Computer Vision [7]–[10], Game theory [11], [12], Speech Recognition [13], Security [14]–[24], Medical diagnosis [25], [26], Statistical Arbitrage [27], Network Anomaly Detection [28]–[32], Learning associations [33], [34], Prediction [35]–[39], Extraction of information [40]–[43], Biometrics [44]–[46], Regression [47], Financial Services [48]–[53] and Classification [54]–[56]. Depending on their perspective on the problem and the approach utilized, scholars have defined machine learning in different ways. However, the underlying definition is nearly identical across the board and revolves around Brett Lantz's definition, who stated that Machine learning is the process of developing computer algorithms for translating data into intelligence [57]. Supervised learning is a broad topic of machine learning that entails learning a data-to-output label mapping. The most popular subtopic in supervised learning is the classification of labeled data [58].

During the previous few decades, classifiers have been intensively studied and analyzed. Classifiers are widely used in many modern applications as key computer technology. Many classifiers exist, such as K-Nearest Neighbor (KNN), Support vector machine (SVM), Naive Bayes (NB), Random Forest (RF), Decision Tree (DT), and many more. In terms of accuracy and time consuming building a trained model, each classifier has advantages and limitations; some are more effective with specific datasets than others, and hence there is no optimal classifier that can perfectly classify all types of data.

The time spent building the trained model, or the test time if the classifier does not provide a trained model, such as the KNN and its variants, is one of the primary concerns with classical classifiers when employed on Big data. Such a long

The associate editor coordinating the review of this manuscript and approving it for publication was Chien-Ming Chen.

time consumed makes the classifier impractical to be used for Big data classification.

In order to hasten the learning process, particularly when training Big data, in this paper, we developed a new classifier based on the idea of Magnetic Force (MF), where we programmatically imitate the MF's work, as inspired by the work of [59]. In this approach, each class is represented by a specific magnet, the iron filings represent the unknown data points that need to be classified. The magnetic field around each magnet/class is formed by the examples belonging to the same class. To calculate the force of each magnet/class on each point in the features space, we apply the Inverse Square Law (ISL). Knowing that the feature space may contain continuous data makes calculating the MFs in an infinity space problematic. Therefore, we opt for discretizing the features space by binning each value in each dimension. Thus, the MF for each class can be calculated for each bin. After building the MF model, which is simply based on the training data, any unknown data point (iron filling) will be attracted to the strongest MF and classified by that particular class.

Formally, we can say: Given a numeric dataset D, with dimensionality d, with n training examples belonging to either class A or Class B, if we can discretize these examples by binning them to b discrete values, then we can generate two Magnets (MA and MB) and store their forces in M, which is b × d matrix. Since the MF is disproportionate to the distance between the magnet and the iron filling, we use ISL, as the nearest iron filings are attracted to either MA or MB more than the furthest. we can form two matrices, one for each magnet/class, or one matrix with two values in each cell. Similarly, this definition is also applied for multi-class datasets.

Obviously, the time complexity of the proposed approach is linear O(n), since it depends mainly on the number of examples found in a training set. Particularly, when $n \gg d$ (number of features) of the training data set. And equally important, the small size of the trained model $= b \times d$, if d is relatively small, then the time and space complexity of the generated model will be O(c), where c is a constant. The ability to classify Big data at such a low level of time and space complexity is critical.

II. LITERATURE REVIEW

Although the KNN classifier is extremely slow and lazy learner [60]–[65], it is still used extensively for Big data classification, because it skips the training process, and save its time, which is extremely long for most of the traditional classifiers. However, it is used with a helper to hasten the test phase: indexing, clustering, sorting, and reducing the data among other reprocesses help speed up the classifier when dealing with Big data.

For example, Hassanat [66] created a new approach to sorting the examples of training data into a binary search tree to speed up Big data classification. This has been carried out by using two methods. The first is referred to as Extreme Points-based binary search tree (EPBST), which identifies local points based on their similarity to these global extremes. And the second referred to as Random-Points-based binary search tree (RPBST), selects the local points randomly. Different experiments on medium and large datasets show reasonable accuracy rates compared to other related methods, including the pure KNN classifier.

Another approach by the same author [67] was proposed to classify Big data using the KNN classifier. The KNN classifier was used in conjunction with inserting training examples into a binary search tree (BST) to speed up the search process, having known that a BST can be searched in logarithmic time. In their work, they examined two methods to sort the training examples. The first is called NBT, which calculates the minimum/maximum criterion measured and rounds it to 0 or 1 for each instance. While the second one, which is called MNBT, inserts each instance in the BST depending on its similarity with the minimum and maximum instances, their experiments show competing accuracy and fast classification compared to the other methods examined.

In a similar work [68], the author created a similar BST, which is used by the KNN classifier to classify Big data. The difference is that this method is based on finding the diameter of a data set, which is then used to sort the BST for training dataset examples. This method has a high potential for classifying Big data and can also be generalized to other applications, particularly when speed is a key factor. The outermost pair of examples could be found for each BST node, and the examples in a particular node are sorted after which the generated BTS could be searched. The examples in this paper are used to classify the test example using the KNN classifier. The results showed the method's efficiency in terms of speed and accuracy compared to the other methods examined. However, compared to the pure KNN, the accuracy rates are not ideal, and it needs more improvements.

The diameter of a dataset used in the previous work was found using methods proposed by A [69]. Where they implemented four simple algorithms to approximate the diameter of a multi-dimensional dataset because most algorithms do not fit well with large values of data dimensions since time complexity grows significantly in most cases. The results of the experiments conducted on different machine learning datasets prove the efficiency of the implemented algorithms. These methods are helpful to find the diameter of a Big dataset, which might be used by machine learning applications. For example, the authors of [70] utilize the diameter concept to solve the imbalanced data problem by synthesizing similar minority examples, using an over-sampling technique based on the furthest neighbour algorithm.

In order to increase the performance of the previous work [66]–[68] in terms of computation time, space consumption, and accuracy, [71] the main contribution is the conversion of the resultant BST to a decision tree. This eliminates the need for the slow KNN and results in a smaller tree, which is useful for memory usage, speeds up both the training and testing phases, and improves the classification accuracy as well. The reported experimental results show that the proposed methods are more efficient in terms of space, speed, and accuracy than [66]–[68] and other methods compared. Indicating that the proposed methods have a lot of room for improvement before they can be used in practice.

The authors of [72] proposed a new approach that improves the instance-based KNN algorithm, their clustering-based strategy was adopted to reduce the computational cost of the KNN algorithm. The performance improved by solving the problem of instances by expanding its size using neural networks to obtain a suitable representation of the classification process. The reported results show better performance compared to the other methods examined.

Another approach [73] proposes a new MapReduce-based approach to classify Big data. The map stage finds the k-nearest neighbors in various data partitions. Following that, the reduce stage computes the exact neighbors from the map stage's list. The proposed approach enables the KNN classifier to scale to any dataset size by adding more nodes as needed. Furthermore, like with the original KNN classifier, this kind of parallel implementation gives the same classification rate. The reported results utilizing a dataset with up to 1 million instances reveal that the proposed approach scales well with Big data.

Another MapReduce-based approach is proposed by [74], The main focus of this proposal is on the test set management, with the goal of maintaining the test set in memory wherever possible. Otherwise, it is partitioned into the smallest number of chunks possible, utilizing MapReduce per chunk and Spark's caching capabilities to reuse the formerly partitioned training data. The reported results utilizing a dataset with up to 11 million examples reveal that the proposed approach scales well with Big data.

A multivariable random decision tree is proposed by [75], the purpose of this method is to speed up the classification process utilizing two partition methods. The first randomly partitions the data, referred to as (MDT1), while the second (MDT2) uses Principal Component Analysis (PCA) for partitioning the data. According to the reported results, both methods allow for short training time and competing accuracy on Big datasets.

For speeding up the KNN classifier, [76] proposed two methods. The first uses random clustering (RC-KNN), and the second, use landmark spectral clustering (LC-KNN). the clustering process is based on the well-known K-means clustering, which is mainly used to split the entire data set into several smaller parts. This makes applying the KNN to test input instances much faster than applying it on the entire dataset. The reported results of both methods LC-KNN, and RC-KNN showed better performance when tested on Big datasets.

In [77], the authors conducted a survey-like study that included the-state-of-the-art techniques and tools used for big data classification. Also, they analyse various approaches that have been followed for big data classification. The advantages and disadvantages of each approach were discussed for different types of machine learning techniques Finally they provided the readers with a survey of many open-source libraries that have been used in big data.

It's also worth highlighting other Big Data classification-related studies, like [78]–[143].

III. THE MAGNETIC FORCE CLASSIFIER (MF)

In this paper, We propose a supervised machine learning Magnetic Force classifier (MF) based on the concept of magnetic force attracting an iron filling. e.g. if we place a piece of iron between two or more magnets, it will be attracted to one of them, which will exert a greater force on it due to the magnet force and the distance from the magnet.

For example, assume that we have three magnets (M1, M2, and M3), an iron filling (F), and a set of arrows representing magnetic forces and directions of attraction towards each magnet, as shown in Figure 1. The blue scale color indicates the presence of varied magnetic strengths in the area; the fainter the hue, the stronger the magnetic force. The highest magnetic force found at each point determines the direction of each arrow, as seen in this diagram. Despite the fact that F is closer to M2 and M3, it is drawn to M1 in this case because M1's force on F is stronger.

We simulate this physical process by representing each class in a training dataset by a magnet with a specific force for each point in the feature space. The training process of the MF classifier ends up generating a trained model that incorporates all of the magnetic forces for each class for each point in the feature space. However, because the feature space is typically defined in terms of real numbers, the model's size would be infinite if each feature were not digitized into discrete values. Binning the training data set allows the MF at each location for each class/magnet to be determined.

We used ISL to properly reflect the magnetic force as being inversely proportional to the distance at each point belonging to each class, with say 1 Tesla added to the force of its magnet/class at that particular point, but it affects all the other locations by adding a fraction of force (FF) as follows

$$FF = \frac{1}{(d+1)^2} \tag{1}$$

where d is the block distance from the current cell.

The output model is a 2D matrix of vectors, where each vector in each cell contains the magnetic force of each class. Table 1 shows the MF trained model of a toy example dataset with two classes/magnets and two features using 10 bins. As also shown in Figure 2.

As demonstrated in Figure 2, larger values exhibit stronger MF for Class 2 (MF2), whereas smaller values exhibit stronger MF for Class 1 (MF1). This means that the class 1 points in our toy example have tiny values for both features, whereas the class 2 points have large values for both features. And this is exactly what happens when we designed our toy example just for the purpose of illustration, as the values of the features of the points belonging to class 1 are in the range of 0 to 5, while those of class 2 are in the range of 4 to 9. As a result, the MF classifier translates the training

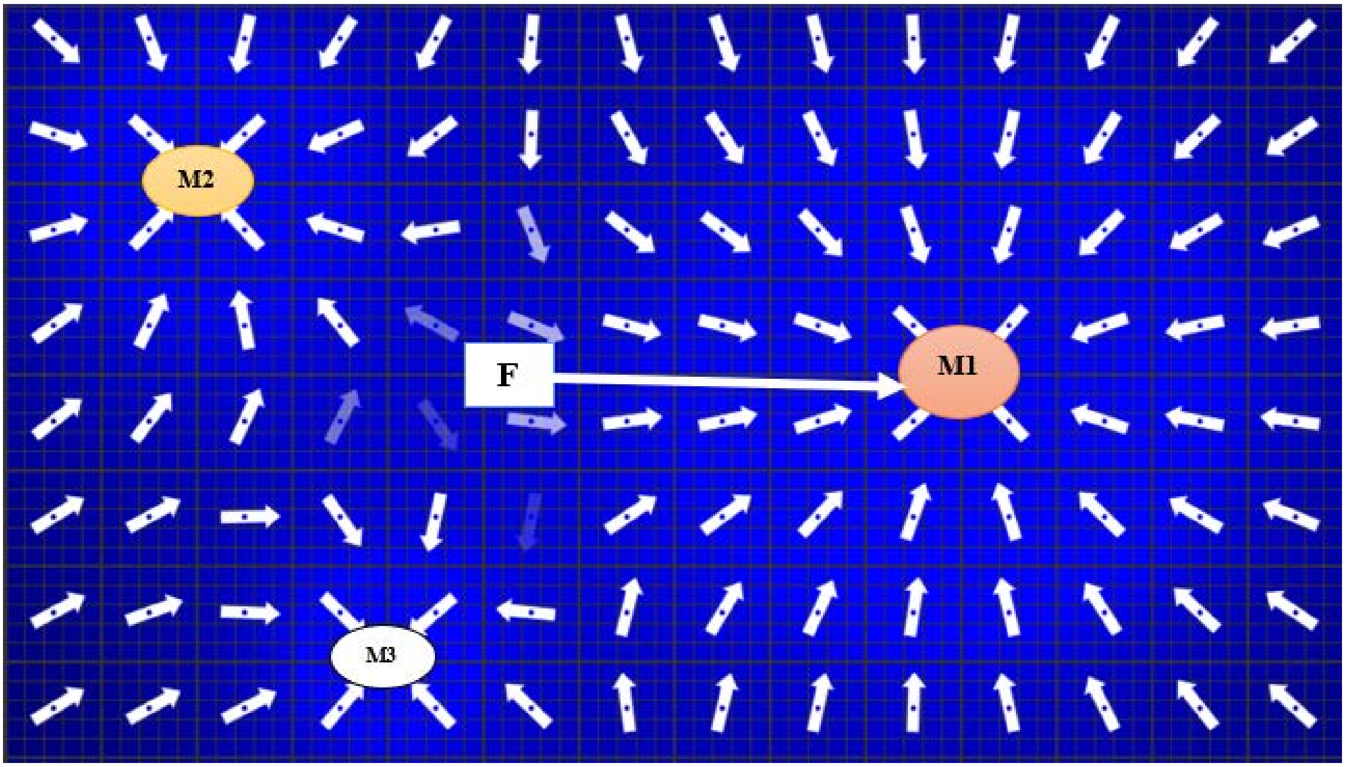


FIGURE 1. Example of 3 magnetic forces (M1, M2, and M3) attempting to attract an iron filling (F).

TABLE 1. The MF trained model of a toy example dataset with two features and two classes.

Feature 1		Feature 2	
MF1	MF2	MF1	MF2
4.20	0.33	2.20	0.36
3.64	0.46	6.92	0.51
6.85	0.67	7.65	0.79
5.63	1.12	6.67	1.45
5.24	2.61	5.12	4.26
4.52	4.89	2.71	4.20
1.56	5.68	1.17	5.22
0.85	7.37	0.70	5.49
0.55	6.31	0.47	6.08
0.39	3.89	0.34	4.68

data into magnetic forces for each class based on the training examples and their features.

During the testing phase, we assume that each tested example is represented by an iron filling, and we assign the test example to the class with the maximum total magnetic force by reading the MF model at the location of the iron filling for each feature, as follows:

$$C = \arg\max_{C_i}^{n} \sum_{k=1}^{m} (MF_k) \tag{2}$$

where C is the predicted class, m is the number of features, MF_k is the magnetic force of feature k, and n is the number of classes.

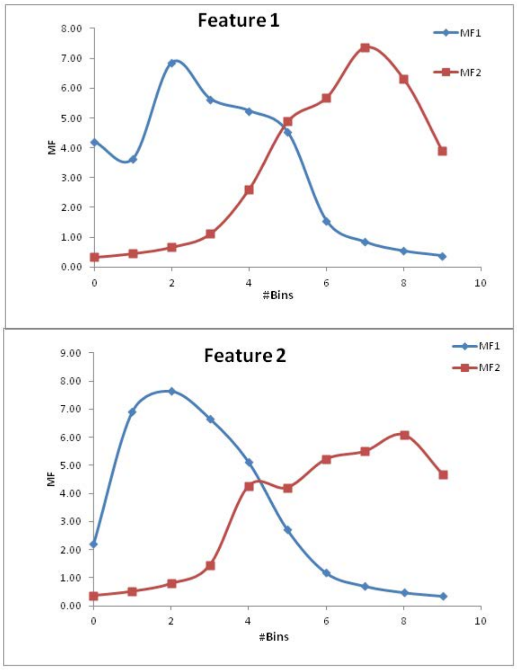


FIGURE 2. Toy example: magnatic force for each feature for each class.

For example, assume we have a point $P_1 = (1, 2)$, according to the toy example model shown in Table 1, MF_1 for $feature_1 = 4.20$ and $MF_1forfeature_2 = 6.92$, then C_1, which represents the MF of $class_1 = 11.12$. Similarly,

Algorithm 1 Training phase

```
Input: Numerical training dataset (train) with C classes, F features, N instances and B Bins
Output: MF as magnetic power values (3D array) Initialize:
  • counter as integer 1D array of size (C)
  • A1, MF as integer 3D arrays of size (B, F-1, C)
  • A2 as double 2D array of size (B, B)
  for feature=0 to F-1 do
      for bin=0 to B do
          for sample = 0 to n do
              if bin == train (sample, feature) then
                  counter (train(sample, feature)) is incremented by 1
              end if
          end for
      end for
      for label=0 to C do
          A1 (bin, feature, label) = counter(label)
          counter (label)=0
      end for
  end for
  for feature = 0 to F do
      for label=0 to C do
          for bin1=0 to B do
              for bin2=0 to B do
                  $A2(bin2, bin1) = \frac{A1(bin1, feature, label)}{((bin2 - bin1) + 1)^2}$
              end for
          end for
          for bin3 = 0 to B do
              for feature2 = 0 to F do
                  $MF(bin3, feature, label) = MF(bin3, feature, label) + A2(bin3, feature2)$
              end for
          end for
      end for
  end for
```

Algorithm 2 Testing phase

```
Input: Numerical testing dataset (test) with C classes, F features, N instances
Output: CM as Confusion matrix
Initialize:
  • counter as 1D integer array of size (C)
  • CM as 2D integer array of size (C,C)
  • max and accuracy as doubles
  for sample = 0 to N do
      for feature in F-1 do
          for label in C do
              $counter(label) = counter(label) + MF(test(sample, feature), feature, label)$
          end for
      end for
      for label2 in C do
          if max< counter (label2) then
              max = counter (label2)
              key = label2
          end if
      end for
      if key == test(sample,F-1) then
          accuracy is incremented by 1
          CM(test(sample, F-1), test(sample, F-1)) is incremented by 1
      else
          CM(test(sample,F-1),key) is incremented by 1
          counter=0
          max=0
      end if
  end for
```

MF_2 for $feature_1 = 0.33$, MF_2 for $feature_2 = 0.51$, then C_2, which represents the MF of $class_2 = 0.84$. And therefore, the predicted class should be $class_1$ according to Equation 2. Both of the training and testing phases are described in algorithms (1) and (2) respectively.

IV. RESULTS AND DISCUSSION

We chose 28 datasets of various sizes to assess the proposed MF classifier: small, medium, and Big datasets. Our implementation is based on these numeric datasets, which were retrieved from the UCI website [144] and the Kaggle website.[1]

Table 2 describes the datasets in terms of number of Instances, number of features, number of classes, and their class imbalance ratio. These datasets differ in terms of the number of examples, features, and classes; this diversity is intended to demonstrate the new classifier's strengths and flaws in terms of accuracy and time consumption when compared to other classifiers.

Because it is difficult to compute an infinite number of different magnetic forces with the proposed MF classifier, it is necessary to scale each feature in the dataset to a discrete range, then build the number of bins depending on this range in order to calculate the magnetic force at each position. Therefore, selecting the right number of bins for MF is critical. We conducted multiple pilot tests utilizing the proposed MF classifier with varying number of bins $2, 3, 4 \ldots, 15$ applied to four (small, medium, and Big) datasets to determine which number of bins is preferable for the MF classifier. Table 3 shows the accuracy of MF for each number of bins.

The number of bins utilized had no effect on Big datasets, as shown in Table 3. This is due to the class imbalance problem, as both Big datasets (SUSY and Poker) are class imbalanced. This is something we'll discuss over later. However, we should highlight that for small and medium-sized datasets (wine and Chinese-minst), the more the number of bins, the higher the accuracy. We should also mention that the accuracy improves from 9 onward. We chose 9 bins for the MF classifier in our experiments as the best option at the present in terms of accuracy and model size, because model size has a considerable impact on memory consumption and testing time.

The proposed MF classifier has been tested and compared to five of the most common classifiers, namely, KNN, SVM, NB, RF, and DT(J48). The comparison focuses on the classification accuracy and time consumed in both the training and testing phases of each classifier. We made these comparisons on the same computer, which has the following specifications:

- Intel(R) Core (TM) i7-9750H CPU @ 2.60GHz
- 16.0 GB of RAM
- 64-bit operating System

[1] https://www.kaggle.com/

TABLE 2. The datasets that were used to assess the MF classifier. CIR is the class imbalance ratio.

ID	Dataset	# samples	# dimensions	# Classes	CIR
D1	Iris	150	4	3	Balanced
D2	Sonar	208	60	2	Imbalanced
D3	Heart failure	299	12	2	Imbalanced
D4	Heart Attack	303	13	2	Imbalanced
D5	Ionosphere	351	34	2	Imbalanced
D6	Vote	399	16	2	Imbalanced
D7	Wholesale	440	7	2	Imbalanced
D8	Monkey	556	17	2	Balanced
D9	Australian	614	42	2	Imbalanced
D10	Balance	625	4	3	Imbalanced
D11	Cancer	683	9	2	Imbalanced
D12	BCW	699	10	2	Imbalanced
D13	Diabetes	768	8	2	Imbalanced
D14	Vehicle	846	18	4	Imbalanced
D15	Parkinson	1040	27	2	Balanced
D16	Chinese-mnist	15000	4	15	Balanced
D17	Wine	3198	12	2	Imbalanced
D18	Waveform21	5000	21	3	Balanced
D19	Phoneme	5404	5	2	Imbalanced
D20	Satimage	6430	36	6	Imbalanced
D21	Letter-recognition	20000	16	26	Balanced
D22	Sensorless_drive_diagnosis	58509	48	11	Balanced
D23	Connect-4.txt	67557	42	3	Imbalanced
D24	VehicleAge	381109	11	2	Imbalanced
D25	Forest (Covtype)	581012	54	7	Imbalanced
D26	Poker	1025010	10	10	Imbalanced
D27	SUSY	5000000	18	2	Imbalanced
D28	HIGGS	11000000	28	2	Imbalanced

TABLE 3. MF accuracy results for each number of bins (2 to 15), when applied on four datasets (small, medium and big).

Datasets \ Bins	2	3	4	5	6	7	8	9	10	11	12	13	14	15
SUSY	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52
Poker	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Chinese-minst	0.19	0.25	0.32	0.4	0.47	0.53	0.59	0.66	0.73	0.79	0.86	0.94	1	1
Wine	0.73	0.85	0.92	0.94	0.95	0.96	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97

- Application used: Weka v 3.8.5 64-bit.

Because the performance of the proposed MF is compared to the aforementioned classifiers using WEKA [145], which is constructed using JAVA computer language, we implemented the MF classifier using Java language too so that the results of the comparisons would be well-comparable.

We used 5-fold cross-validation to evaluate the MF classifier in a range of trials, comparing the results to baseline classifiers and assessing the accuracy and time spent in the training and testing phases. Table 4 shows the times spent by all classifiers in both the training and testing phases for all of the 5-fold training/testing experiments. As can be noted from these results, the MF classifier required significantly less time than the other classifiers.

However, the time was comparable with small datasets because most classifiers are quick on short datasets, but using baseline classifiers on Big datasets like Poker, SUSY, and HIGGS took over 4 weeks to build and test their models, so we halted their experiments and were unable to add their results to Table. 4.

The evaluated baseline classifiers are inefficient when used on Big data for two reasons: 1) they take an unreasonable amount of time to build and test, and 2) they require a lot of memory, whereas the MF classifier took roughly 10 minutes to build and test its model (5 times since 5-fold) for the Biggest dataset (Higgs).

The complexity of the MF classifier is determined by a number of factors, including the number of samples (n), features or dimensions (d), classes (c), and bins employed (b). So the size of the generated MF model $= b \times d \times c$, however, if $n \gg d$, which is the case in most datasets, including the ones used in this study, then d may be treated as a constant, and hence the space complexity becomes $= O(constant)$, which is asymptotically $= O(1)$. Similarly, the time complexity of creating the MF model $= n \times d$ becomes $= O(n)$, i.e. linear time, for the same reason. The proposed MF classier's efficiency demonstrated by the findings presented

TABLE 4. Time (ms) consumed by each classifier for training and testing phases.

id	MF	KNN	SVM	NB	RF	DT
D1	16	20	30	10	50	10
D2	15	20	30	10	90	20
D3	17	10	10	10	60	10
D4	17	10	20	10	40	140
D5	18	20	30	10	60	10
D6	17	10	10	10	40	10
D7	18	10	10	10	60	10
D8	22	10	20	10	70	10
D9	21	10	190	10	140	10
D10	17	10	10	10	70	10
D11	17	10	10	10	40	10
D12	19	10	10	10	60	10
D13	19	10	10	10	140	10
D14	21	10	30	10	190	10
D15	27	20	60	10	150	10
D16	94	1162	561	150	1650	96
D17	61	30	90	20	420	30
D18	80	320	110	20	1560	90
D19	65	140	110	10	1030	10
D20	92	390	220	50	1890	70
D21	140	2750	3620	200	6000	160
D22	380	61350	24150	940	37630	2140
D23	337	103000	559160	805	52660	2490
D24	1000	175503160	2301900	1300	311140	11410
D25	7271	6600200	97214000	11480	1626230	77444
D26	12110	–	–	–	–	–
D27	55175	–	–	–	–	–
D28	612045	–	–	–	–	–

in Table 4 is due to its minimal complexity in time and space.

In terms of classification accuracy, Table 5 displays the accuracy of categorizing 25 datasets (small and medium) using MF and baseline classifiers, leaving the results of Big datasets to be compared to current Big data classifiers, as we were unable to obtain their results owing to time and memory constraints.

Large magnets have larger magnetic fields, and hence a larger attraction force on iron fillings in real life. The same physics applies to the proposed MF classifier, we expect that examples with large numbers will dominate the MF learning process on the account of other examples belonging to minority classes. Because a high number of instances from the same class have a strong magnetic force, hence unclassified examples are incorrectly classified to be belonging to the majority class, even though they are closer to another minority example.

Because the majority of the datasets used in this study are class imbalanced, in order to investigate the impact of class imbalance on our MF classifier, we used the MF classifier on the datasets as-is, then balanced them, and compared the results before and after the balancing process. For simplicity, we balanced all datasets by randomly dropping extra examples from the majority class [146].

As can be observed from the classification accuracy results in Table 5, and as expected, after balancing, the accuracy increased for the majority of datasets. However, after balancing, the accuracy results of some datasets are close to or less than the accuracy before balancing. This is due to the data balancing method we utilized; because we used Random Under-sampling, where instances are discarded at random, these instances may include important information for better learning.

The impact of class imbalance on MF may be examined further by comparing the confusion matrices before and after balancing the data to better understand how the MF classifier behaves while classifying balanced and imbalanced data.

We used four datasets to calculate the confusion matrix before and after balancing just for simplicity. A closer examination of the confusion matrices in Figure 3 offers

TABLE 5. Comparison of MF classification accuracy to baseline classifiers, with MF results obtained solely before and after data balancing.

id	MF before balancing	MF after balancing	KNN	SVM	NB	RF	DT
D1	0.96	0.96	0.97	0.96	0.96	0.966	0.966
D2	0.8	0.82	0.88	0.85	0.66	0.857	0.619
D3	0.65	0.71	0.7	0.86	0.8	0.833	0.866
D4	0.75	0.8	0.7	0.83	0.86	0.852	0.836
D5	0.64	0.84	0.86	0.84	0.8	0.857	0.814
D6	0.95	0.93	0.93	0.96	0.95	0.975	0.96
D7	0.73	0.84	0.89	0.85	0.86	0.909	0.909
D8	0.65	0.65	1	0.75	0.75	1	0.936
D9	0.71	0.83	0.81	0.86	0.84	0.89	0.87
D10	0.93	0.86	0.78	0.86	0.86	0.75	0.728
D11	0.79	0.89	0.97	0.99	0.97	0.985	0.94
D12	0.77	0.9	0.95	0.96	0.95	0.97	0.93
D13	0.73	0.74	0.74	0.79	0.77	0.82	0.78
D14	0.6	0.61	0.69	0.76	0.44	0.75	0.733
D15	0.76	0.76	0.88	0.9	0.97	1	1
D16	1	1	1	1	1	1	1
D17	0.77	0.97	0.96	0.92	0.81	0.98	0.95
D18	0.76	0.76	0.75	0.88	0.81	0.849	0.784
D19	0.72	0.82	0.89	0.77	0.73	0.913	0.82
D20	0.69	0.75	0.91	0.86	0.8	0.93	0.84
D21	0.51	0.51	0.95	0.82	0.64	0.964	0.844
D22	0.64	0.64	0.98	0.88	0.71	0.99	0.98
D23	0.65	0.7	0.789	0.65	0.59	0.81	0.75
D24	0.88	0.82	0.87	0.87	0.77	0.872	0.87
D25	0.49	0.48	0.94	0.72	0.63	0.95	0.924
D26	0.5	–	0.5	0.5	0.5	–	0.66
D27	0.88	–	–	–	–	–	–
D28	0.52	–	–	–	–	–	–

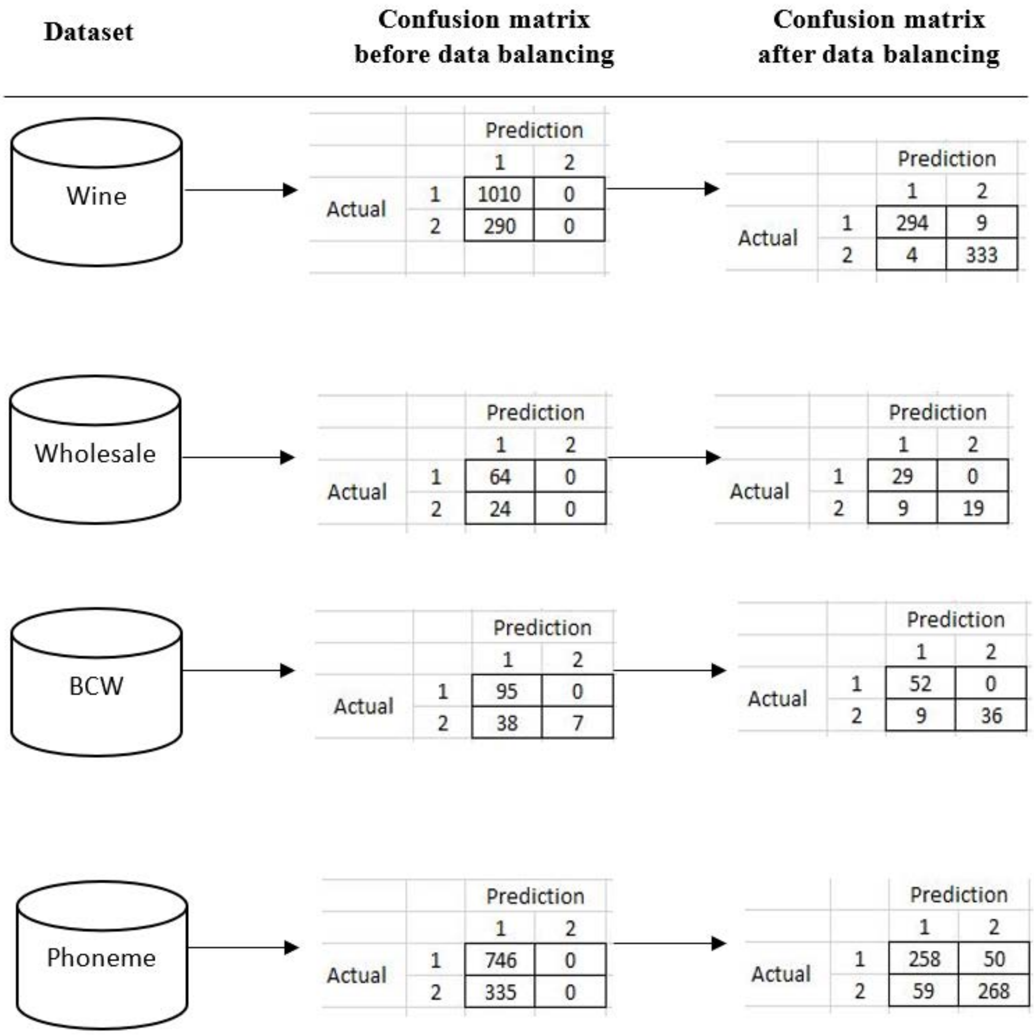
TABLE 6. Comparison of MF classification accuracy/training time(ms) to big data classifiers.

Datasets Methods	Poker	SUSY	HIGGS
MF	0.5 / **750**	**0.88 / 4728**	0.52 / **163513**
NBT	0.46 / 229624	0.59 / 157615	0.52 / 345090
MNBT	0.48 / 170782	0.71 / 330117	0.59 1215761
MDT1	0.5 / 12100	0.73 / 567000	0.58 8330000
MDT2	**0.54** / 10900	0.75 / 345000	**0.60** 2450000
RPBST	0.486 / 66629	0.709 / 617419	0.588 427656
EPBST	0.475 / 147167	0.791 / 1096631	0.585 844253
KNN	0.502 / 105475006	0.694 / 3258848	-
MR-KNN	0.502 / 804456	0.694 / 12367966	-
KNN-Is	0.502 / 102938	0.694 / 1900039	-
FPBST	0.53 / 35516	0.75 / 255481	-

further information about the MF model's evaluation; for instance, before balancing, the majority of the examples were classified as belonging to the majority class, but after balancing, the classification results were significantly improved, as shown by the confusion matrices in Figure 3.

Based on this finding, we conclude that the proposed MF classifier has a fundamental limitation when learning from class imbalanced data in its current form.

Previous research has employed a variety of methods to speed up big data classification. We compared the proposed MF classifier's efficiency in terms of accuracy and time spent building and testing the model with the findings of prior studies, which include EPBST, RPBST [66], FPBST [68], NBT, MNBT [67], Iterative MapReduce-based approach for kNN (MR-KNN) [73], Iterative Spark-based design of the kNN classifier (KNN-IS) [74], Randomly partitioned multivariate decision tree (MDT1) and the PCA-partitioned


Dataset
Confusion matrix before data balancing
Confusion matrix after data balancing
Wine
Wholesale
BCW
Phoneme

Wine, before data balancing:

		Prediction	
		1	2
Actual	1	1010	0
	2	290	0

Wine, after data balancing:

		Prediction	
		1	2
Actual	1	294	9
	2	4	333

Wholesale, before data balancing:

		Prediction	
		1	2
Actual	1	64	0
	2	24	0

Wholesale, after data balancing:

		Prediction	
		1	2
Actual	1	29	0
	2	9	19

BCW, before data balancing:

		Prediction	
		1	2
Actual	1	95	0
	2	38	7

BCW, after data balancing:

		Prediction	
		1	2
Actual	1	52	0
	2	9	36

Phoneme, before data balancing:

		Prediction	
		1	2
Actual	1	746	0
	2	335	0

Phoneme, after data balancing:

		Prediction	
		1	2
Actual	1	258	50
	2	59	268

FIGURE 3. MF Confusion matrices for four dataset, before and after data balancing.

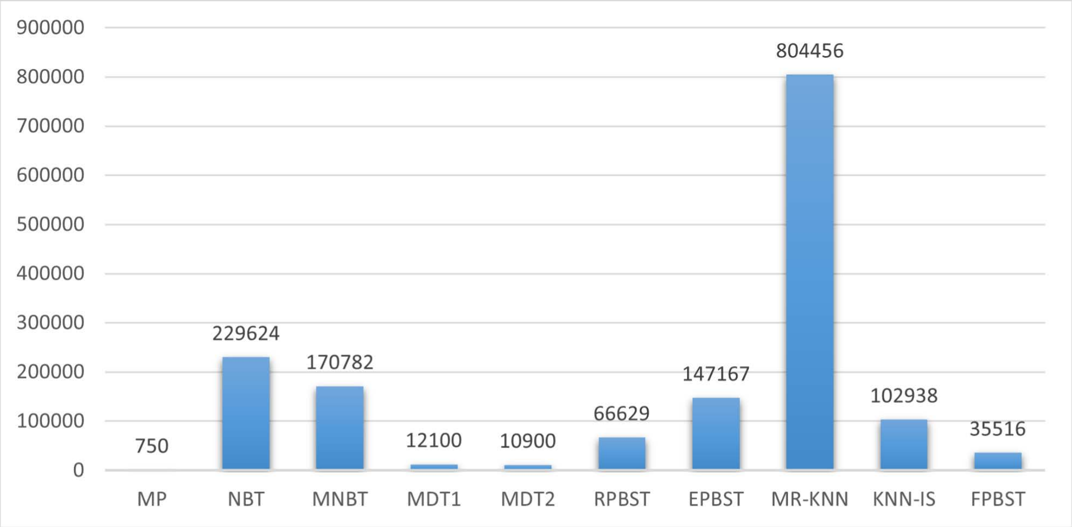


FIGURE 4. Time consumed by the evaluated classifiers on Poker Big dataset.

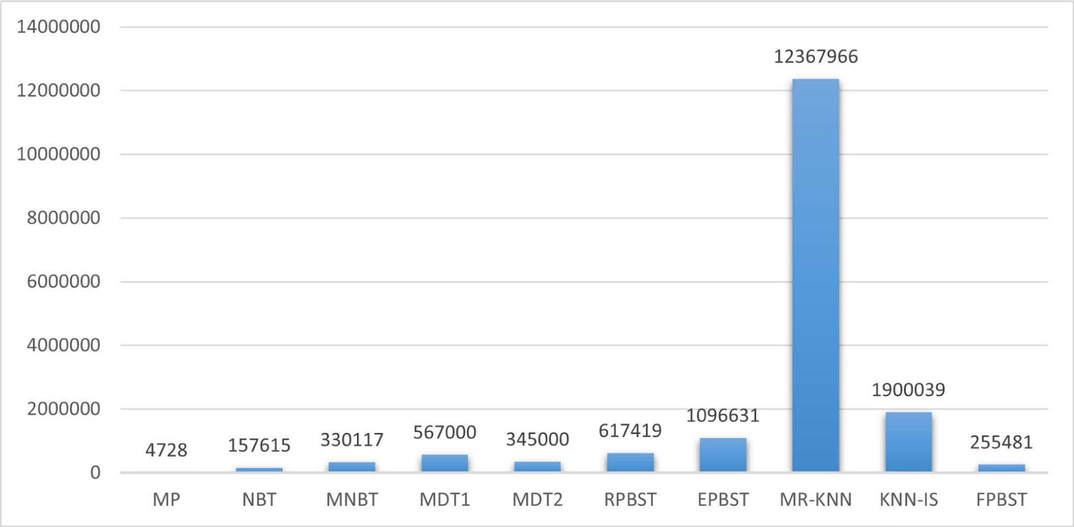


FIGURE 5. Time consumed by the evaluated classifiers on SUSY Big dataset.

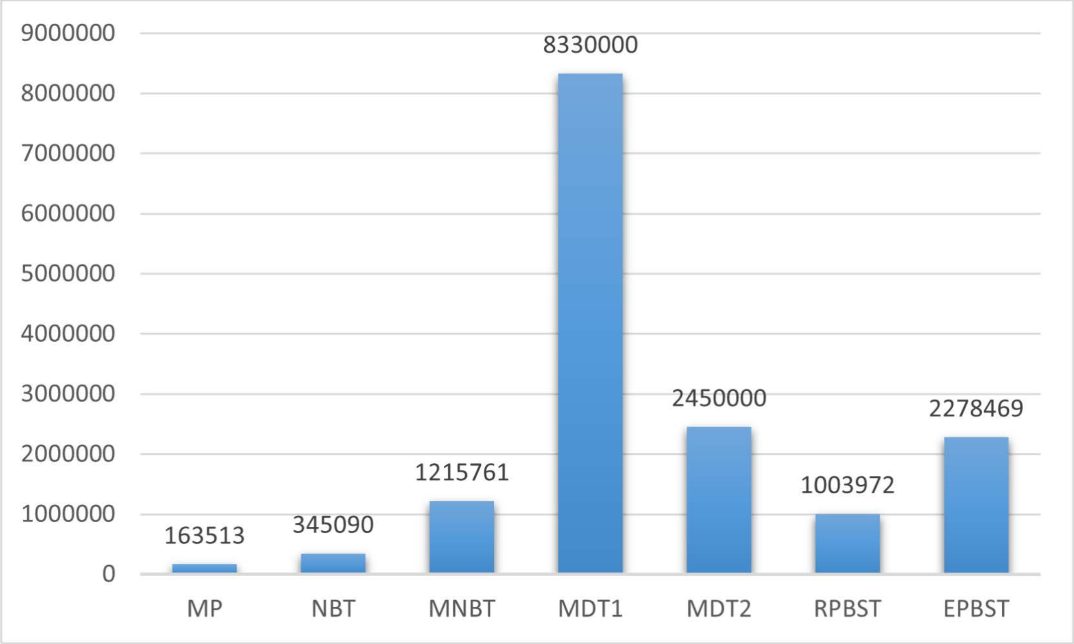


FIGURE 6. Time consumed by the evaluated classifiers on Higgs Big dataset.

multivariate decision tree (MDT2) [75]. These methods each published findings on one or more of the three Big datasets D26, D27, and D28, also known as Poker, SUSY, and HIGGS.

Using 5-fold cross-validation, we compared the classification results achieved by the MF classifier to EPBST, RPBST, FPBST, NBT, MNBT on the three Big datasets. All of these methods rely on constructing a BST model during the training phase, which makes the test example search significantly faster than sequential search. The other Big data classifiers are also included in the comparison as shown in Table 6.

It is worth noting that the time consumed in Table 6 is for training and testing phases for some methods and for training only for others, depending on the published results; likewise, the training/testing ratio is not always 5-fold cross-validation; it is holdout with test to train ratios ranging from 19% to 90%.

As shown in Table 6, the MF classifier consumes significantly less time, proving its high speed and efficiency when applied to Big data. The MF Accuracy results are comparable to those of the other Big data classifiers but significantly outperform them all when used on the SUSY Big dataset. Figures 4, 5 and 6, depict the significantly less time consumed by the MF classifier compared to the other Big data classifiers evaluated on the three Big datasets used.

The time comparison was based on time reported in the literature, but we genuinely think it is valid because most of the previous methods compared used machines with high specifications and multiple CPUs, so as to be able to deal with Big data. For example, EPBST, RPBST, FPBST, NBT, and MNBT, all used Azure high-performance computing virtual machine with 16 CPUs and 32 GB RAM, [66], [68], and [67]. While the proposed MF classifier was evaluated on a standard laptop with the modest specifications listed at the opening of this section.

V. CONCLUSION

Inspired by the power of nature, and the work of Hassanat et. al [59], we propose the MF classifier, which calculates the magnetic force at each discrete point in the feature space based on the number of points belonging to a certain class/magnet. The forces measured by various magnets/classes- which are recorded in the trained MF model- are then utilized to classify unknown samples.

We employed 28 small, medium, and big benchmark datasets to evaluate the proposed classifier, and compared the classification results and time consumed by the training and testing phases to a number of popular classifiers and a number of Big data classifiers.

The experimental findings reveal that when compared to the other classifiers, the proposed MF classifier achieves comparable classification accuracy. More importantly, we found that the proposed classifier is significantly faster than all of the other classifiers assessed, especially when applied to Big datasets and hence could be a viable choice for structured Big data classification.

The proposed classifier, however, has two major limitations, according to the results of the experiments: 1) Deciding on the optimal number of bins for the MF model, which is determined by a variety of parameters such as the size of the output model, classification accuracy, and data type. And 2) The current version of the MF model is sensitive to data that is skewed by class. Both of these issues will be addressed in future work.

REFERENCES

[1] I. K. Raharjana, D. Siahaan, and C. Fatichah, "User stories and natural language processing: A systematic literature review," *IEEE Access*, vol. 9, pp. 53811–53826, 2021.

[2] M. Alghamdi and W. Teahan, "Experimental evaluation of Arabic OCR systems," *PSU Res. Rev.*, vol. 1, no. 3, pp. 229–241, Nov. 2017.

[3] H. E. Salman, A.-D. Seriai, and C. Dony, "Feature-level change impact analysis using formal concept analysis," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 25, no. 1, pp. 69–92, Feb. 2015.

[4] H. Eyal Salman, M. Hammad, A.-D. Seriai, and A. Al-Sbou, "Semantic clustering of functional requirements using agglomerative hierarchical clustering," *Information*, vol. 9, no. 9, p. 222, Sep. 2018.

[5] H. E. Salman, "Identification multi-level frequent usage patterns from Apis," *J. Syst. Softw.*, vol. 130, pp. 42–56, Aug. 2017.

[6] A. B. A. Hassanat and G. A. Altarawneh, "Rule-and dictionary-based solution for variations in written Arabic names in social networks, big data, accounting systems and large databases," *Res. J. Appl. Sci., Eng. Technol.*, vol. 8, no. 14, pp. 1630–1638, Oct. 2014.

[7] A. S. Tarawneh, C. Celik, A. B. Hassanat, and D. Chetverikov, "Detailed investigation of deep features with sparse representation and dimensionality reduction in cbir: A comparative study," *Intell. Data Anal.*, vol. 24, no. 1, pp. 47–68, 2020.

[8] A. S. Tarawneh, A. B. Hassanat, C. Celik, D. Chetverikov, M. S. Rahman, and C. Verma, "Deep face image retrieval: A comparative study with dictionary learning," in *Proc. 10th Int. Conf. Inf. Commun. Syst. (ICICS)*, Jun. 2019, pp. 185–192.

[9] A. B. A. Hassanat, M. Alkasassbeh, M. Al-awadi, and E. A. A. Alhasanat, "Colour-based lips segmentation method using artificial neural networks," in *Proc. 6th Int. Conf. Inf. Commun. Syst. (ICICS)*, Apr. 2015, pp. 188–193.

[10] A. Hassanat and A. S. Tarawneh, "Fusion of color and statistic features for enhancing content-based image retrieval systems," *J. Theor. Appl. Inf. Technol.*, vol. 88, no. 3, pp. 644–655, 2016.

[11] A. de Voogt, A. B. A. Hassanat, and M. B. Alhasanat, "The history and distribution of tāb: A survey of Petra's gaming boards," *J. Near Eastern Stud.*, vol. 76, no. 1, pp. 93–101, Apr. 2017.

[12] A. B. Hassanat, G. Altarawneh, A. S. Tarawneh, H. Faris, M. B. Alhasanat, A. de Voogt, B. Al-Rawashdeh, M. Alshamaileh, and S. V. B. Prasath, "On computerizing the ancient game of tāb," *Int. J. Gaming Comput.-Mediated Simul.*, vol. 10, no. 3, pp. 20–40, Jul. 2018.

[13] H. Luo, S. Zhang, M. Lei, and L. Xie, "Simplified self-attention for transformer-based end-to-end speech recognition," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, Jan. 2021, pp. 75–81.

[14] M. Almseidin, I. Piller, M. Al-Kasassbeh, and S. Kovacs, "Fuzzy automaton as a detection mechanism for the multi-step attack," *Int. J. Adv. Sci., Eng. Inf. Technol.*, vol. 9, no. 2, pp. 575–586, 2019.

[15] M. Almseidin, A. A. Zuraiq, M. Alkasassbeh, and N. Alnidami, "Phishing detection based on machine learning and feature selection method," *Int. J. Interact. Mobile Technol.*, vol. 13, no. 12, pp. 171–183, 2019.

[16] M. Al-kasassbeh and T. Khairallah, "Winning tactics with DNS tunnelling," *Netw. Secur.*, vol. 2019, no. 12, pp. 12–19, Dec. 2019.

[17] M. Alkasassbeh, S. Mohammed, M. Alauthman, and A. Almomani, "Feature selection using a machine learning to classify a malware," in *Handbook of computer networks and cyber Security*. Cham, Switzerland: Springer, 2020, pp. 889–904.

[18] A. Abadleh, S. Han, S. J. Hyun, B. Lee, and M. Kim, "Construction of indoor floor plan and localization," *Wireless Netw.*, vol. 22, no. 1, pp. 175–191, Jan. 2016.

[19] A. Abadleh, B. M. Al-Mahadeen, R. M. AlNaimat, and O. Lasassmeh, "Noise segmentation for step detection and distance estimation using smartphone sensor data," *Wireless Netw.*, vol. 27, no. 4, pp. 2337–2346, May 2021.

[20] A. Abadleh, E. Al-Hawari, E. Alkafaween, and H. Al-Sawalqah, "Step detection algorithm for accurate distance estimation using dynamic step length," in *Proc. 18th IEEE Int. Conf. Mobile Data Manage. (MDM)*, May 2017, pp. 324–327.

[21] A. Aljaafreh, K. Alawasa, S. Alja'afreh, and A. Abadleh, "Fuzzy inference system for speed bumps detection using smart phone accelerometer sensor," *J. Telecommun., Electron. Comput. Eng.*, vol. 9, pp. 33–136, Sep. 2017.

[22] E. Hamadaqa, A. Abadleh, A. Mars, and W. Adi, "Highly secured implantable medical devices," in *Proc. Int. Conf. Innov. Inf. Technol. (IIT)*, Nov. 2018, pp. 7–12.

[23] A. Mars, A. Abadleh, and W. Adi, "Operator and manufacturer independent D2D private link for future 5G networks," in *Proc. IEEE INFOCOM Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2019, pp. 1–6.

[24] S. Mulhem, A. Abadleh, and W. Adi, "Accelerometer-based joint user-device clone-resistant identity," in *Proc. 2nd World Conf. Smart Trends Syst., Secur. Sustainability (WorldS4)*, Oct. 2018, pp. 230–237.

[25] K. A. Bhavsar, A. Abugabah, J. Singla, A. A. AlZubi, and A. K. Bashir, "A comprehensive review on medical diagnosis using machine learning," *Comput., Mater. Continua*, vol. 67, no. 2, p. 1997, 2021.

[26] I. El Naqa and M. J. Murphy, "What is machine learning?" in *Machine Learning in Radiation Oncology*. Cham, Switzerland: Springer, 2015, pp. 3–11.

[27] S. M. Carta, S. Consoli, A. S. Podda, D. R. Recupero, and M. M. Stanciu, "Ensembling and dynamic asset selection for risk-controlled statistical arbitrage," *IEEE Access*, vol. 9, pp. 29942–29959, 2021.

[28] M. Alkasassbeh, "A novel hybrid method for network anomaly detection based on traffic prediction and change point detection," *J. Comput. Sci.*, vol. 14, no. 2, pp. 153–162, Feb. 2018.

[29] G. Al-Naymat, M. Alkasassbeh, and E. Al-Hawari, "Using machine learning methods for detecting network anomalies within snmp-mib dataset," *Int. J. Wireless Mobile Comput.*, vol. 15, no. 1, pp. 67–76, 2018.

[30] M. Almseidin, M. Al-Kasassbeh, and S. Kovacs, "Detecting slow port scan using fuzzy rule interpolation," in *Proc. 2nd Int. Conf. new Trends Comput. Sci. (ICTCS)*, Oct. 2019, pp. 33–38.

[31] N. Nasri, S. Mnasri, and T. Val, "3D node deployment strategies prediction in wireless sensors network," *Int. J. Electron.*, vol. 107, no. 5, pp. 808–838, May 2020.

[32] S. Tlili, S. Mnasri, and T. Val, "A multi-objective gray wolf algorithm for routing in IoT collection networks with real experiments," in *Proc. Nat. Comput. Colleges Conf. (NCCC)*, Mar. 2021, pp. 1–5.

[33] T. Ozcan and K. Gunasti, "How associations between products and numbers in brand names affect consumer attitudes: Introducing multi-context numbers," *J. Brand Manage.*, vol. 26, no. 2, pp. 176–194, Mar. 2019.

[34] W. Abdallah and T. Val, "Genetic-Voronoi algorithm for coverage of IoT data collection networks," in *Proc. 30th Int. Conf. Comput. Theory Appl. (ICCTA)*, Dec. 2020, pp. 1–6.

[35] W. Höpken, T. Eberle, M. Fuchs, and M. Lexhagen, "Improving tourist arrival prediction: A big data and artificial neural network approach," *J. Travel Res.*, vol. 60, no. 5, pp. 998–1017, 2021.

[36] P. Narloch, A. Hassanat, A. S. Tarawneh, H. Anysz, J. Kotowski, and K. Almohammadi, "Predicting compressive strength of cement-stabilized rammed Earth based on SEM images using computer vision and deep learning," *Appl. Sci.*, vol. 9, no. 23, p. 5131, Nov. 2019.

[37] S. Mnasri, N. Nasri, A. Van Den Bossche, and T. Val, "A comparative analysis with validation of NSGA-III and MOEA/D in resolving the 3D indoor redeployment problem in DL-IoT," in *Proc. Int. Conf. Internet Things, Embedded Syst. Commun. (IINTEC)*, Oct. 2017, pp. 15–20.

[38] S. Mnasri, A. Van Den Bossche, N. Nasri, and T. Val, "The 3D redeployment of nodes in wireless sensor networks with real testbed prototyping," in *Proc. Int. Conf. Ad-Hoc Netw. Wireless*. Cham, Switzerland: Springer, 2017, pp. 18–24.

[39] S. Mnasri, N. Nasri, M. Alrashidi, A. van den Bossche, and T. Val, "IoT networks 3D deployment using hybrid many-objective optimization algorithms," *J. Heuristics*, vol. 26, no. 5, pp. 663–709, Oct. 2020.

[40] J. Qu and C. Wangtragulsang, "Information extraction for thai celebrities from free text," *Sci. Technol. Asia*, pp. 64–83, 2021.

[41] S. Mnasri, N. Nasri, A. Van Den Bossche, and T. Val, "3D indoor redeployment in IoT collection networks: A real prototyping using a hybrid PI-NSGA-III-VF," in *Proc. 14th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2018, pp. 780–785.

[42] W. Abdallah, S. Mnasri, N. Nasri, and T. Val, "Emergent IoT wireless technologies beyond the year 2020: A comprehensive comparative analysis," in *Proc. Int. Conf. Comput. Inf. Technol. (ICCIT)*, Sep. 2020, pp. 1–5.

[43] S. Mnasri, F. Abbes, K. Zidi, and K. Ghedira, "A multi-objective hybrid BCRC-NSGAII algorithm to solve the VRPTW," in *Proc. 13th Int. Conf. Hybrid Intell. Syst. (HIS)*, Dec. 2013, pp. 60–65.

[44] M. Z. Al-Shamaileh, A. B. Hassanat, A. S. Tarawneh, M. S. Rahman, C. Celik, and M. Jawthari, "New online/offline text-dependent Arabic handwriting dataset for writer authentication and identification," in *Proc. 10th Int. Conf. Inf. Commun. Syst. (ICICS)*, Jun. 2019, pp. 116–121.

[45] A. I. Al-Btoush, M. A. Abbadi, A. B. Hassanat, A. S. Tarawneh, A. Hasanat, and V. B. S. Prasath, "New features for eye-tracking systems: Preliminary results," in *Proc. 10th Int. Conf. Inf. Commun. Syst. (ICICS)*, Jun. 2019, pp. 179–184.

[46] A. B. A. Hassanat, "On identifying terrorists using their victory signs," *Data Sci. J.*, vol. 17, p. 27, Oct. 2018.

[47] M. J. Awan, M. S. M. Rahim, H. Nobanee, A. Munawar, A. Yasin, and A. M. Z. Azlanmz, "Social media and stock market prediction: A big data approach," *Comput., Mater. Continua*, vol. 67, no. 2, pp. 2569–2583, 2021.

[48] B. G. Buchanan and D. Wright, "The impact of machine learning on U.K. financial services," *Oxford Rev. Econ. Policy*, vol. 37, no. 3, pp. 537–563, Sep. 2021.

[49] O. Masters, H. Hunt, E. Steffinlongo, J. Crawford, and F. Bergamaschi, "Towards a homomorphic machine learning big data pipeline for the financial services sector," *IACR Cryptol. ePrint Arch.*, vol. 1, no. 1, p. 1113, 2019.

[50] S. Mnasri, N. Nasri, A. van den Bossche, and T. Val, "A new multi-agent particle swarm algorithm based on birds accents for the 3D indoor deployment problem," *ISA Trans.*, vol. 91, pp. 262–280, Aug. 2019.

[51] S. Mnasri, A. V. D. Bossche, N. Narsi, and T. Val, "The 3D deployment multi-objective problem in mobile WSN: Optimizing coverage and localization," *Int. Res. J. Innov. Eng.*, vol. 1, no. 5, pp. 1–14, 2015.

[52] S. Mnasri, N. Nasri, A. Van Den Bossche, and T. Val, "A hybrid ant-genetic algorithm to solve a real deployment problem: A case study with experimental validation," in *Proc. Int. Conf. Ad-Hoc Netw. Wireless*. Cham, Switzerland: Springer, 2017, pp. 367–381.

[53] S. Mnasri, N. Nasri, and T. Val, "The 3D indoor deployment in DL-IoT with experimental validation using a particle swarm algorithm based on the dialects of songs," in *Proc. 14th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2018, pp. 928–933.

[54] A. S. Tarawneh, D. Chetverikov, C. Verma, and A. B. Hassanat, "Stability and reduction of statistical features for image classification and retrieval: Preliminary results," in *Proc. 9th Int. Conf. Inf. Commun. Syst. (ICICS)*, Apr. 2018, pp. 117–121.

[55] A. B. Hassanat, V. S. Prasath, B. M. Al-Mahadeen, and S. M. M. Alhasanat, "Classification and gender recognition from veiled-faces," *Int. J. Biometrics*, vol. 9, no. 4, pp. 347–364, 2017.

[56] A. S. Tarawneh, A. B. Hassanat, D. Chetverikov, I. Lendak, and C. Verma, "Invoice classification using deep features and machine learning techniques," in *Proc. IEEE Jordan Int. Joint Conf. Electr. Eng. Inf. Technol. (JEEIT)*, Apr. 2019, pp. 855–859.

[57] B. Lantz, *Machine Learning With R: Expert Techniques for Predictive Modeling*. Birmingham, U.K.: Packt Publishing LTD, 2019.

[58] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," *Emerg. Artif. Intell. Appl. Comput. Eng.*, vol. 160, pp. 3–24, Oct. 2007.

[59] A. B. A. Hassanat, V. B. S. Prasath, M. Al-Kasassbeh, A. S. Tarawneh, and A. J. Al-Shamailh, "Magnetic energy-based feature extraction for low-quality fingerprint images," *Signal, Image Video Process.*, vol. 12, no. 8, pp. 1471–1478, Nov. 2018.

[60] S. Muthaiyah and V. A. Singh, "Bone cancer survivability prognosis with KNN and genetic algorithms," in *Concepts and Real-Time Applications of Deep Learning*. Cham, Switzerland: Springer, 2021, pp. 123–134.

[61] E. Ebrahimi and M. Shourian, "River flow prediction using dynamic method for selecting and prioritizing K-nearest neighbors based on data features," *J. Hydrolog. Eng.*, vol. 25, no. 5, May 2020, Art. no. 04020010.

[62] M. Yang, H. Wang, Y. Jiang, X. Lu, Z. Xu, and G. Sun, "GECA proposed ensemble–KNN method for improved monthly runoff forecasting," *Water Resour. Manage.*, vol. 34, no. 2, pp. 849–863, Jan. 2020.

[63] M. Mahfouz, "RFKNN: Rough-fuzzy KNN for big data classification," *Int. J. Adv. Res. Comput. Sci.*, vol. 9, p. 274, Mar. 2018.

[64] W. Hou, D. Li, C. Xu, H. Zhang, and T. Li, "An advanced k nearest neighbor classification algorithm based on KD-tree," in *Proc. IEEE Int. Conf. Saf. Produce Informatization (IICSPI)*, Dec. 2018, pp. 902–905.

[65] R. D. Ravipati and M. Abualkibash, "A survey on different machine learning algorithms and weak classifiers based on KDD and NSL-KDD datasets," *Int. J. Artif. Intell. Appl.*, vol. 10, no. 3, pp. 1–11, May 2019.

[66] A. B. A. Hassanat, "Two-point-based binary search trees for accelerating big data classification using KNN," *PLoS ONE*, vol. 13, no. 11, Nov. 2018, Art. no. e0207772.

[67] A. Hassanat, "Norm-based binary search trees for speeding up KNN big data classification," *Computers*, vol. 7, no. 4, p. 54, Oct. 2018.

[68] A. B. A. Hassanat, "Furthest-pair-based binary search tree for speeding big data classification using K-nearest neighbors," *Big Data*, vol. 6, no. 3, pp. 225–235, Sep. 2018.

[69] A. Hassanat, "Greedy algorithms for approximating the diameter of machine learning datasets in multidimensional Euclidean space: Experimental results," *Adv. Distrib. Comput. Artif. Intell. J.*, vol. 7, no. 3, pp. 15–30, Sep. 2018.

[70] A. S. Tarawneh, A. B. A. Hassanat, K. Almohammadi, D. Chetverikov, and C. Bellinger, "SMOTEFUNA: Synthetic minority over-sampling technique based on furthest neighbour algorithm," *IEEE Access*, vol. 8, pp. 59069–59082, 2020.

[71] A. Hassanat, "Furthest-pair-based decision trees: Experimental results on big data classification," *Information*, vol. 9, no. 11, p. 284, Nov. 2018.

[72] A.-J. Gallego, J. Calvo-Zaragoza, J. J. Valero-Mas, and J. R. Rico-Juan, "Clustering-based k-nearest neighbor classification for large-scale data with neural codes representation," *Pattern Recognit.*, vol. 74, pp. 531–543, Feb. 2018.

[73] J. Maillo, I. Triguero, and F. Herrera, "A MapReduce-based k-nearest neighbor approach for big data classification," in *Proc. IEEE Trustcom/BigDataSE/ISPA*, Aug. 2015, pp. 167–172.

[74] J. Maillo, S. Ramírez, I. Triguero, and F. Herrera, "kNN-IS: An Iterative Spark-based design of the K-nearest Neighbors classifier for big data," *Knowl.-Based Syst.*, vol. 117, pp. 3–15, Feb. 2016.

[75] F. Wang, Q. Wang, F. Nie, W. Yu, and R. Wang, "Efficient tree classifiers for large scale datasets," *Neurocomputing*, vol. 284, pp. 70–79, Apr. 2018.

[76] Z. Deng, X. Zhu, D. Cheng, M. Zong, and S. Zhang, "Efficient kNN classification algorithm for big data," *Neurocomputing*, vol. 195, pp. 143–148, Jun. 2016.

[77] P. K. D. Pramanik, S. Pal, M. Mukhopadhyay, and S. P. Singh, "Big data classification: Techniques and tools," in *Applications of Big Data in Healthcare*. Amsterdam, The Netherlands: Elsevier, 2021, pp. 1–43.

[78] D. García-Gil, J. Luengo, S. García, and F. Herrera, "Enabling smart data: Noise filtering in big data classification," *Inf. Sci.*, vol. 479, pp. 135–152, Apr. 2019.

[79] N. A. Al-Thanoon, Z. Y. Algamal, and O. S. Qasim, "Feature selection based on a crow search algorithm for big data classification," *Chemometric Intell. Lab. Syst.*, vol. 212, May 2021, Art. no. 104288.

[80] W. C. Sleeman IV and B. Krawczyk, "Multi-class imbalanced big data classification on spark," *Knowl.-Based Syst.*, vol. 212, Jan. 2021, Art. no. 106598.

[81] S. M. Mujeeb, R. P. Sam, and K. Madhavi, "Adaptive exponential bat algorithm and deep learning for big data classification," *Sādhanā*, vol. 46, no. 1, pp. 1–15, Dec. 2021.

[82] M. Juez-Gil, Á. Arnaiz-González, J. J. Rodríguez, and C. García-Osorio, "Experimental evaluation of ensemble classifiers for imbalance in big data," *Appl. Soft Comput.*, vol. 108, Sep. 2021, Art. no. 107447.

[83] J. A. R. Cedeño, C. F. Leopoldo, H. Neira-Molina, Y. J. García-López, R. Morales-Ortega, and H. Combita-Niño, "Big data classification using fuzzy logical concepts for paddy yield prediction," *Rev. Int. Geograph. Educ. Online*, vol. 11, no. 5, pp. 4483–4490, 2021.

[84] N. P. Jayasri and R. Aruna, "Big data analytics in health care by data mining and classification techniques," *ICT Exp.*, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405959521000849, doi: 10.1016/j.icte.2021.07.001.

[85] M. Juez-Gil, Á. Arnaiz-González, J. J. Rodríguez, C. López-Nozal, and C. García-Osorio, "Approx-SMOTE: Fast SMOTE for big data on apache spark," *Neurocomputing*, vol. 464, pp. 432–437, Nov. 2021.

[86] D. C. Asogwa, S. O. Anigbogu, I. E. Onyenwe, and F. A. Sani, "Text classification using hybrid machine learning algorithms on big data," *Int. J. Trend Res. Develop.*, vol. 6, no. 5, pp. 127–134, 2021.

[87] R. T. Selvi and I. Muthulakshmi, "An optimal artificial neural network based big data application for heart disease diagnosis and classification model," *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 6, pp. 6129–6139, Jun. 2021.

[88] I. Hababeh, A. Gharaibeh, S. Nofal, and I. Khalil, "An integrated methodology for big data classification and security for improving cloud systems data mobility," *IEEE Access*, vol. 7, pp. 9153–9163, 2019.

[89] C. Banchhor and N. Srinivasu, "Integrating cuckoo search-grey wolf optimization and correlative naive Bayes classifier with map reduce model for big data classification," *Data Knowl. Eng.*, vol. 127, May 2020, Art. no. 101788.

[90] A. Rghioui, J. Lloret, and A. Oumnad, "Big data classification and Internet of Things in healthcare," *Int. J. E-Health Med. Commun.*, vol. 11, no. 2, pp. 20–37, Apr. 2020.

[91] J. Maillo, I. Triguero, and F. Herrera, "Redundancy and complexity metrics for big data classification: Towards smart data," *IEEE Access*, vol. 8, pp. 87918–87928, 2020.

[92] W.-D. Xie and X. Cheng, "Imbalanced big data classification based on virtual reality in cloud computing," *Multimedia Tools Appl.*, vol. 79, nos. 23–24, pp. 16403–16420, Jun. 2020.

[93] C. Jiang and Y. Li, "Health big data classification using improved radial basis function neural network and nearest neighbor propagation algorithm," *IEEE Access*, vol. 7, pp. 176782–176789, 2019.

[94] S. M. Mujeeb, R. Praveen Sam, and K. Madhavi, "Energy- and trust-aware secure routing algorithm for big data classification using MapReduce framework in IoT networks," *Int. J. Model., Simul., Sci. Comput.*, vol. 12, no. 2, Apr. 2021, Art. no. 2150010.

[95] D. Ahamad, M. M. Akhtar, and S. AlamHameed, "Optimisation algorithm-based recurrent neural network for big data classification," *Int. J. Intell. Inf. Database Syst.*, vol. 1, no. 1, p. 1, 2020.

[96] M. S. Mohammed, P. S. Rachapudy, and M. Kasa, "Big data classification with optimization driven MapReduce framework," *Int. J. Knowledge-based Intell. Eng. Syst.*, vol. 25, no. 2, pp. 173–183, Jul. 2021.

[97] L.-H. Yang, J. Liu, Y.-M. Wang, and L. Martinez, "A micro-extended belief rule-based system for big data multiclass classification problems," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 51, no. 1, pp. 420–440, Jan. 2021.

[98] F. Abukhodair, W. Alsaggaf, A. T. Jamal, S. Abdel-Khalek, and R. F. Mansour, "An intelligent Metaheuristic binary pigeon optimization-based feature selection and big data classification in a MapReduce environment," *Mathematics*, vol. 9, no. 20, p. 2627, Oct. 2021.

[99] S. K. Punia, M. Kumar, T. Stephan, G. G. Deverajan, and R. Patan, "Performance analysis of machine learning algorithms for big data classification: ML and AI-based algorithms for big data analysis," *Int. J. E-Health Med. Commun.*, vol. 12, no. 4, pp. 60–75, Jul. 2021.

[100] F. BenSaid and A. M. Alimi, "Online feature selection system for big data classification based on multi-objective automated negotiation," *Pattern Recognit.*, vol. 110, Feb. 2021, Art. no. 107629.

[101] I. M. S. Ali and M. Balakrishnan, "Population and global search improved squirrel search algorithm for feature selection in big data classification," *Int. J. Intell. Eng. Syst.*, vol. 14, no. 4, pp. 177–189, Aug. 2021.

[102] C. Banchhor and N. Srinivasu, "Analysis of Bayesian optimization algorithms for big data classification based on map reduce framework," *J. Big Data*, vol. 8, no. 1, pp. 1–19, Dec. 2021.

[103] M. Lu, "Analysis of various techniques for solving the problem of big data classification," *Int. Rev. Model. Simulations (IREMOS)*, vol. 14, no. 4, p. 242, Aug. 2021.

[104] M. J. Basgall, M. Naiouf, and A. Fernández, "FDR2-BD: A fast data reduction recommendation tool for tabular big data classification problems," *Electronics*, vol. 10, no. 15, p. 1757, Jul. 2021.

[105] A. Jp, "Rider chicken optimization algorithm-based recurrent neural network for big data classification in spark architecture," *Comput. J.*, May 2021.

[106] A. V. Brahmane and C. B. Krishna, "Rider chaotic biography optimization-driven deep stacked auto-encoder for big data classification using spark architecture: Rider chaotic biography optimization," *Int. J. Web Services Res.*, vol. 18, no. 3, pp. 42–62, Jul. 2021.

[107] H. Kadkhodaei, A. M. E. Moghadam, and M. Dehghan, "Big data classification using heterogeneous ensemble classifiers in apache spark based on MapReduce paradigm," *Expert Syst. Appl.*, vol. 183, Nov. 2021, Art. no. 115369.

[108] Z. Cai, J. Wang, and M. Ma, "The performance evaluation of big data-driven modulation classification in complex environment," *IEEE Access*, vol. 9, pp. 26313–26322, 2021.

[109] A. K. Dubey, A. Kumar, and R. Agrawal, "An efficient ACO-PSO-based framework for data classification and preprocessing in big data," *Evol. Intell.*, vol. 14, no. 2, pp. 909–922, Jun. 2021.

[110] S. Kamaruddin and V. Ravi, "EGRNN++ and PNN++ : Parallel and distributed neural networks for big data regression and classification," *Social Netw. Comput. Sci.*, vol. 2, no. 2, pp. 1–19, Apr. 2021.

[111] X. Gu, P. Angelov, and Z. Zhao, "Self-organizing fuzzy inference ensemble system for big streaming data classification," *Knowl.-Based Syst.*, vol. 218, Apr. 2021, Art. no. 106870.

[112] M. Ramachandran, R. Patan, A. Kumar, S. Hosseini, and A. H. Gandomi, "Mutual informative MapReduce and minimum quadrangle classification for brain tumor big data," *IEEE Trans. Eng. Manag.*, early access, May 2021, doi: 10.1109/TEM.2021.3073018.

[113] D. K. Jain, P. Boyapati, J. Venkatesh, and M. Prakash, "An intelligent cognitive-inspired computing with big data analytics framework for sentiment analysis and classification," *Inf. Process. Manage.*, vol. 59, no. 1, Jan. 2022, Art. no. 102758.

[114] N. Seliya, A. Abdollah Zadeh, and T. M. Khoshgoftaar, "A literature review on one-class classification and its potential applications in big data," *J. Big Data*, vol. 8, no. 1, pp. 1–31, Dec. 2021.

[115] P. Y. Wang, "Design of distributed multidimensional big data classification system based on differential equation," in *Proc. Int. Conf. Advanced Hybrid Inf. Process.*, vol. 347, 2021, pp. 389–398.

[116] E. Agrillo, F. Filipponi, A. Pezzarossa, L. Casella, D. Smiraglia, A. Orasi, F. Attorre, and A. Taramelli, "Earth observation and biodiversity big data for forest habitat types classification and mapping," *Remote Sens.*, vol. 13, no. 7, p. 1231, Mar. 2021.

[117] A. Shokrzade, M. Ramezani, F. A. Tab, and M. A. Mohammad, "A novel extreme learning machine based kNN classification method for dealing with big data," *Expert Syst. Appl.*, vol. 183, Nov. 2021, Art. no. 115293.

[118] C. Gong, Z.-G. Su, P.-H. Wang, Q. Wang, and Y. You, "Evidential instance selection for K-nearest neighbor classification of big data," *Int. J. Approx. Reasoning*, vol. 138, pp. 123–144, Nov. 2021.

[119] S. K. Hussin, S. M. Abdelmageid, A. Alkhalil, Y. M. Omar, M. I. Marie, and R. A. Ramadan, "Handling imbalance classification virtual screening big data using machine learning algorithms," *Complexity*, vol. 2021, pp. 1–15, Jan. 2021.

[120] A. K. Jaiswal, P. Tiwari, S. Garg, and M. S. Hossain, "Entity-aware capsule network for multi-class classification of big data: A deep learning approach," *Future Gener. Comput. Syst.*, vol. 117, pp. 1–11, Apr. 2021.

[121] R. Sujitha and V. Seenivasagam, "Classification of lung cancer stages with machine learning over big data healthcare framework," *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 5, pp. 5639–5649, May 2021.

[122] H. J. Pushparani and N. J. Goldena, "Reliable distributed fuzzy discretizer for associative classification of big data," *Int. J. Inf. Retr. Res.*, vol. 12, no. 1, pp. 1–13, Jan. 2022.

[123] P. G. Lavanya, K. Kouser, and M. Suresha, "Effective feature representation using symbolic approach for classification and clustering of big data," *Expert Syst. Appl.*, vol. 173, Jul. 2021, Art. no. 114658.

[124] H. Zhang, T. Guo, and X. Su, "Application of big data technology in the impact of tourism E-commerce on tourism planning," *Complexity*, vol. 2021, pp. 1–10, May 2021.

[125] J.-S. Kim and J. Kim, "A study on deep learning model-based object classification for big data environment," *J. Softw. Assessment Valuation*, vol. 17, no. 1, pp. 59–66, Jun. 2021.

[126] H. Yang and J. Wang, "Research on online evaluation method of MOOC teaching quality based on decision tree-based big data classification," *Int. J. Continuing Eng. Educ. Life-Long Learn.*, vol. 1, no. 1, p. 1, 2023.

[127] A. Rana, S. Gautam, P. Mann, and N. Tyagi, "Classification for big data tools and its challenges and issues," *SSRN Electron. J.*, pp. 1–8, 2021, doi: 10.2139/ssrn.3746970.

[128] V. VandanaKolisetty and D. S. Rajput, "Integration and classification approach based on probabilistic semantic association for big data," *Complex Intell. Syst.*, pp. 1–14, 2021, doi: 10.1007/s40747-021-00548-x.

[129] G. Gupta and K. Lakhwani, "An enhanced intelligent classification approach to improve the encryption of big data," in *Proc. IOP Conf. Ser., Mater. Sci. Eng.*, vol. 1049, 2021, Art. no. 012008.

[130] S. Hong, R. Zuo, H. Hu, Y. Xiong, and Z. Wang, "Magnetite geochemical big data: Dataset construction and application in genetic classification of ore deposits," *Earth Sci. Frontiers*, vol. 28, p. 87, May 2021.

[131] F. Khan, B. V. V. S. Prasad, S. A. Syed, I. Ashraf, and L. K. Ramasamy, "An efficient, ensemble-based classification framework for big medical data," *Big Data*, 2021, doi: 10.1089/big.2021.0132.

[132] S. K. Sahu, D. P. Mohapatra, J. K. Rout, K. S. Sahoo, and A. K. Luhach, "An ensemble-based scalable approach for intrusion detection using big data framework," *Big Data*, vol. 9, no. 4, pp. 303–321, Aug. 2021.

[133] H. Chang and Y. Cui, "Image classification algorithm based on big data and multilabel learning of improved convolutional neural network," *Wireless Commun. Mobile Comput.*, vol. 2021, pp. 1–11, Sep. 2021.

[134] S. K. Ramachandran and P. Manikandan, "An efficient ALO-based ensemble classification algorithm for medical big data processing," *Int. J. Med. Eng. Informat.*, vol. 13, no. 1, p. 54, 2021.

[135] W. Alzyadat, A. AlHroob, I. H. Almukahel, M. Muhairat, M. Abdallah, and A. Althunibat, "Big data, classification, clustering and generate rules: An inevitably intertwined for prediction," in *Proc. Int. Conf. Inf. Technol. (ICIT)*, Jul. 2021, pp. 149–155.

[136] Y. Yuan, "Big data mining and artificial intelligence based classification algorithm," in *Proc. Int. Conf. Appl. Intell. Syst. Multi-Modal Inf. Anal.*, 2021, pp. 871–875.

[137] A. Kundaliya and P. Juyal, "Advance deep learning technique for big data classification in IDS environment," in *Proc. 3rd Int. Conf. Inventive Res. Comput. Appl. (ICIRCA)*, Sep. 2021, pp. 1744–1748.

[138] Y. Y. Gao, J. Xiang, Y. N. Tang, M. He, and W. Li, "Research on big data classification algorithm of disease gene detection based on complex network technology," in *Proc. Int. Conf. Adv. Hybrid Inf. Process.*, vol. 347, 2021, pp. 306–319.

[139] P. Bedi, S. B. Goyal, R. Sharma, D. K. Yadav, and M. Sharma, "Smart model for big data classification using deep learning in wireless body area networks," in *Micro-Electronics and Telecommunication Engineering*. Singapore: Springer, 2021, pp. 215–224.

[140] H. Amazal, M. Ramdani, and M. Kissi, "A parallel global tfidf feature selection using Hadoop for big data text classification," in *Advances on Smart and Soft Computing*. Singapore: Springer, 2021, pp. 107–117.

[141] R. Jha, V. Bhattacharjee, and A. Mustafi, "Classification of big data using spark framework," in *Proc. 4th Int. Conf. Microelectron., Comput. Commun. Syst.*, vol. 673, 2021, pp. 847–854.

[142] N. Albarrak, H. Alsanousi, I. Moulitsas, and S. Filippone, "Using big data to compare classification models for household credit rating in kuwait," in *Proc. 6th Int. Congr. Inf. Commun. Technol.*, vol. 216, 2022, pp. 609–618.

[143] I. Pintye, E. Kail, P. Kacsuk, and R. Lovas, "Big data and machine learning framework for clouds and its usage for text classification," *Concurrency Comput., Pract. Exper.*, vol. 33, no. 19, Oct. 2021, Art. no. e6164.

[144] A. Asuncion and D. H. Newman. (2007). UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences. Accessed: Jan. 26, 2022. [Online]. Available: http://archive.ics.uci.edu/ml

[145] F. Eibe, M. A. Hall, and I. H. Witten, *The Weka Workbench. Online Appendix for Data Mining: Practical Machine Learning Tools and Techniques*. San Mateo, CA, USA: Morgan Kaufmann, 2016.

[146] J. Brownlee, *Data Preparation for Machine Learning: Data Cleaning, Feature Selection, and Data Transforms in Python*. Montpelier, VT, USA: Machine Learning Mastery, 2020.

AHMAD B. HASSANAT (Member, IEEE) was born in Jordan. He received the B.S. degree in computer science from Mutah University, Jordan, in 1995, the M.S. degree in computer science from Al al-Bayt University, Jordan, in 2004, and the Ph.D. degree in computer science from the University of Buckingham, U.K., in 2010. He has been a Faculty Member with the Department of Information Technology, Mutah University, since 2010. He is currently doing a sabbatical with the University of Tabuk, Saudi Arabia. His main research interests include computer vision, machine learning, big data, and pattern recognition.

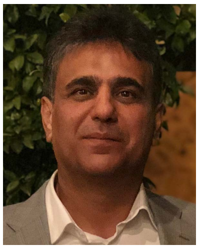
HASAN N. ALI was born in Iraq. He received the B.S. degree in computer science from Tikrit University, Iraq, in 2015, and the master's degree in computer science from Mutah University, majoring in artificial intelligence. He works as a Programmer at the Computer Center, Tikrit University. His main research interests include machine learning, big data, and data mining.

AHMAD S. TARAWNEH was born in Karak, Jordan. He received the B.S. and M.S. degrees in computer science from Mutah University, in 2013 and 2015, respectively, and the Ph.D. degree in computer science from Eötvös Loránd University (ELTE), Budapest, Hungary. He worked on the project of EFOP (image and video processing), which is sponsored by the Hungarian Government and co-financed by the European Social Fund. His main research interests include computer vision, deep learning, machine learning, and data mining.

MALEK ALRASHIDI received the B.Sc. degree in computer science from Taibah University, Saudi Arabia, in 2008, the M.Sc. degree from Newcastle University, U.K., in 2011, and the Ph.D. degree from the School of Intelligence Environment Research Group (IEG) and the Immersive Education Laboratory (iEL). He is currently an Assistant Professor at the University of Tabuk, Saudi Arabia. His main research interests include mixed, augmented, and virtual reality (MR/AR/VR), artificial intelligence (AI) ambient intelligence (AmI), the Internet-of-Things (IoT), cyber-physical systems (CPS), intelligent environments, computer-support collaborative work (CSCW), technology-enhanced learning (TEL), and human–computer interaction (HCI).

MANSOOR ALGHAMDI received the M.Sc. degree from the University of Otago, New Zealand, in 2011, and the Ph.D. degree from the School of Computer Science, Bangor University, U.K. He is currently an Assistant Professor with the Department of Computer Science, University of Tabuk. His main research interests include natural language processing, machine learning, text analytics in English and Arabic, virtual reality, and psychological aspects of mixed reality.

GHADA AWAD ALTARAWNEH received the Ph.D. degree in accounting from the University of Buckingham, U.K., in 2011. She is currently an Associate Professor with Mutah University. Her main interests include managerial accounting, auditing, and business intelligence.



MOHAMMAD ALI ABBADI received the Ph.D. degree in computer science from George Washington University, in 2000. He is currently a Professor with the Department of Computer Science, Faculty of Information Technology, Mutah University. His research interests include multimedia, networks, blended and e-learning, data compression, and image processing.